

# State Elimination - The SDMLib Solution

Alexander Weidt & Albert Zündorf

University Kassel – Software Engineering Research Group

# SDMLib

- **Started March 2012**
- **Successor to Fujaba**
- **Textual (Java-Based) (Graphical tool coming soon)**
- **Lightweight modeling library**
- **contains tools for Graph Transformation**

# Algorithm

## Algorithm 2:

### **convertToGTG**

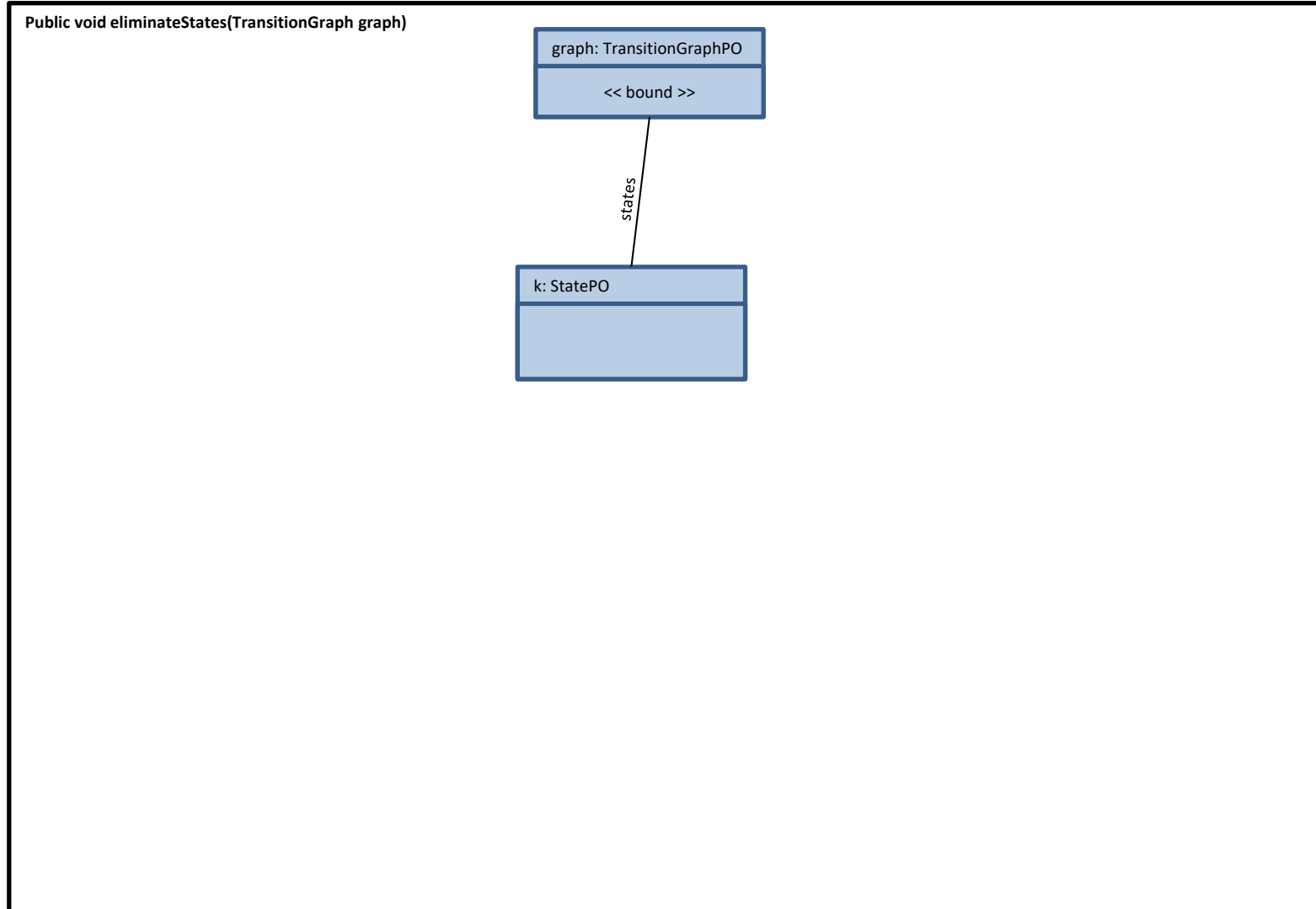
Input: A simple automaton A.

Output: An equivalent GTG A' with only the initial state and the final state.

1. for k:A.getStates() do
2.     newTransitions←[]
3.     if k is not initial or final then
4.         for p,q: A.getStates() do
5.             if p!=k and q!=k then
6.                 pq←getExpressionForElimination(k,p,q,A)
7.                 newTransitions.add(pq)
8.     remove k and all its incoming and outgoing transition from A
9.     A.add(newTransitions)

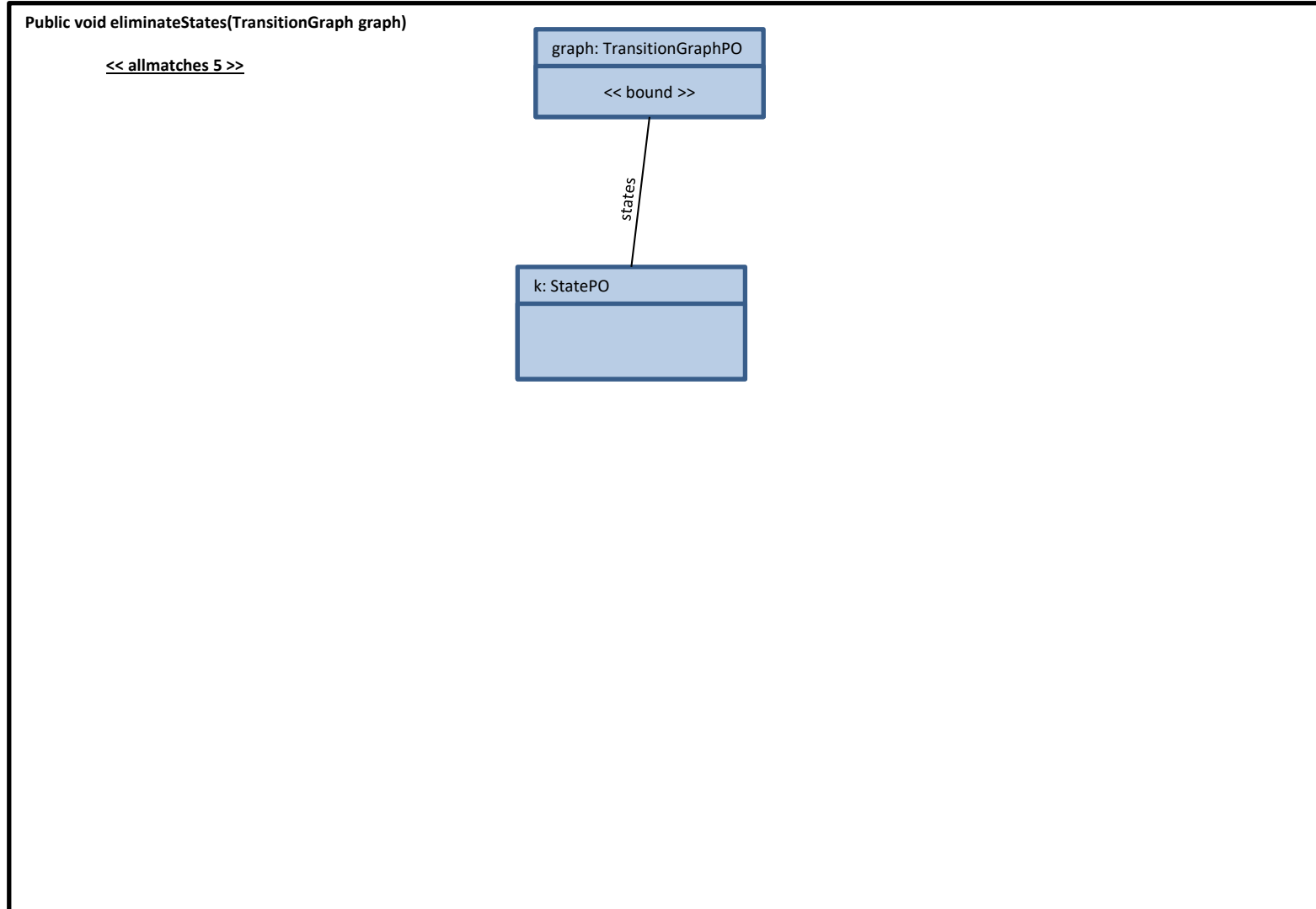
# Pattern

k:A.getStates()



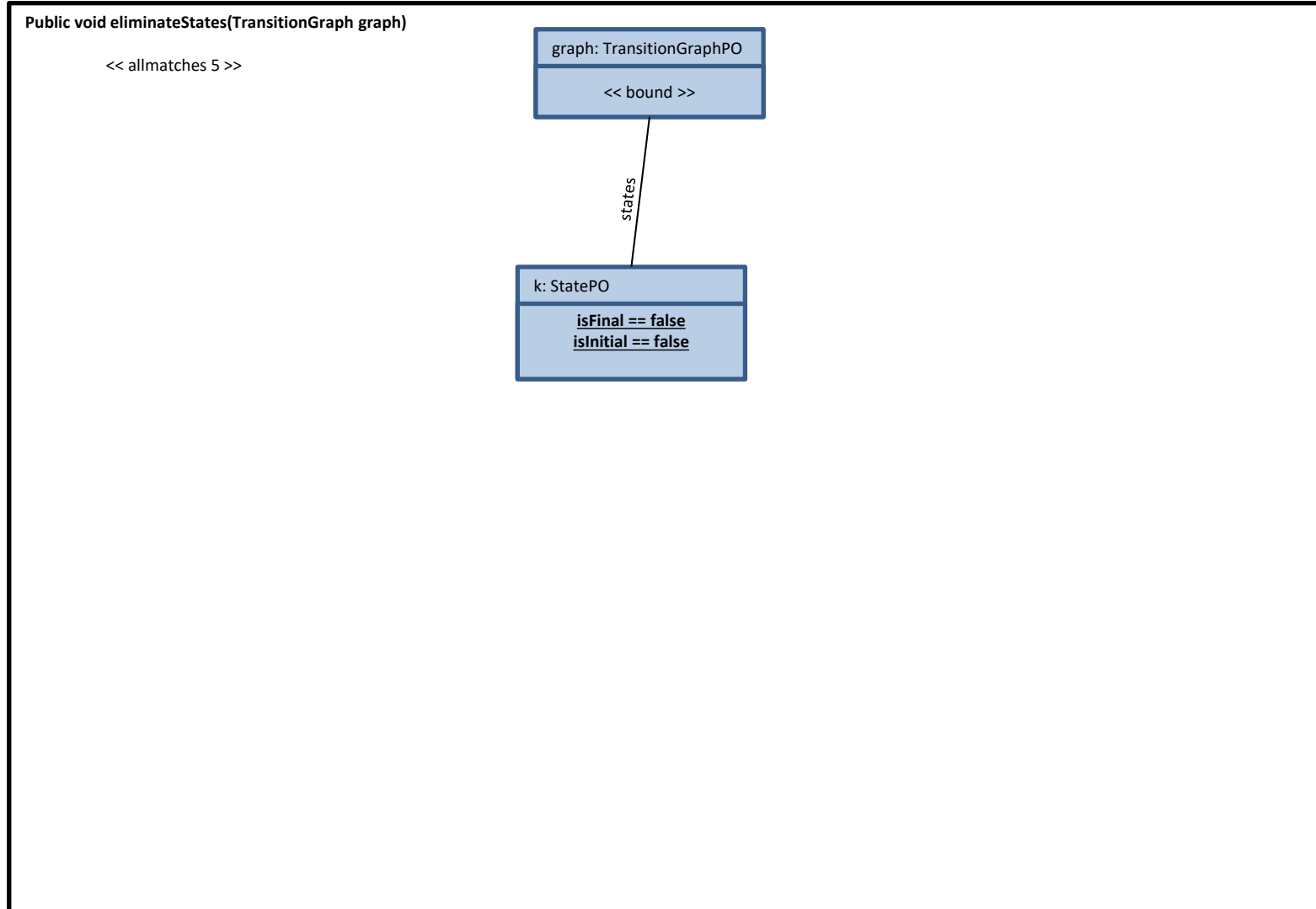
# Pattern

for k:A.getStates() do



# Pattern

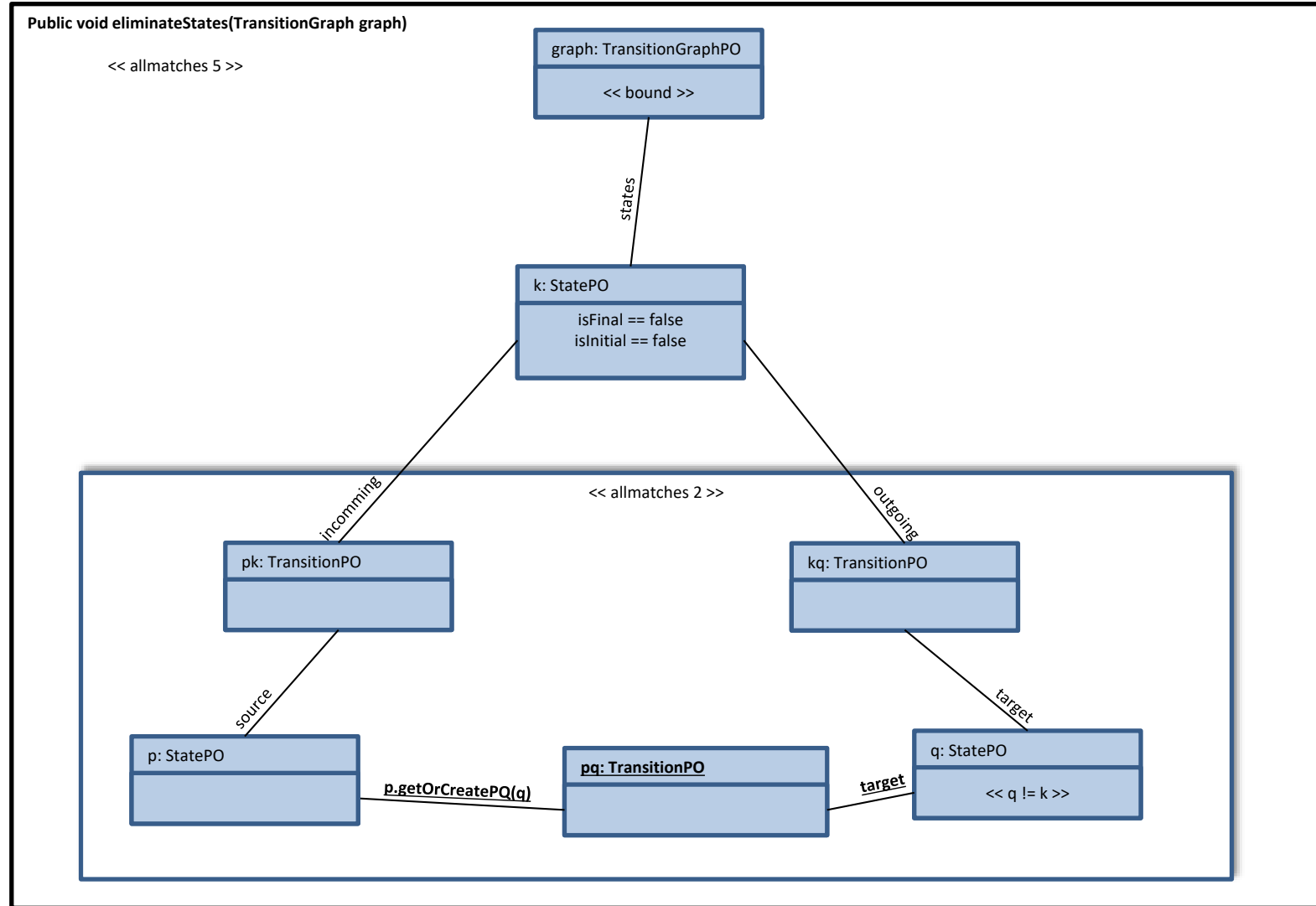
if k is not initial or final then





# Pattern

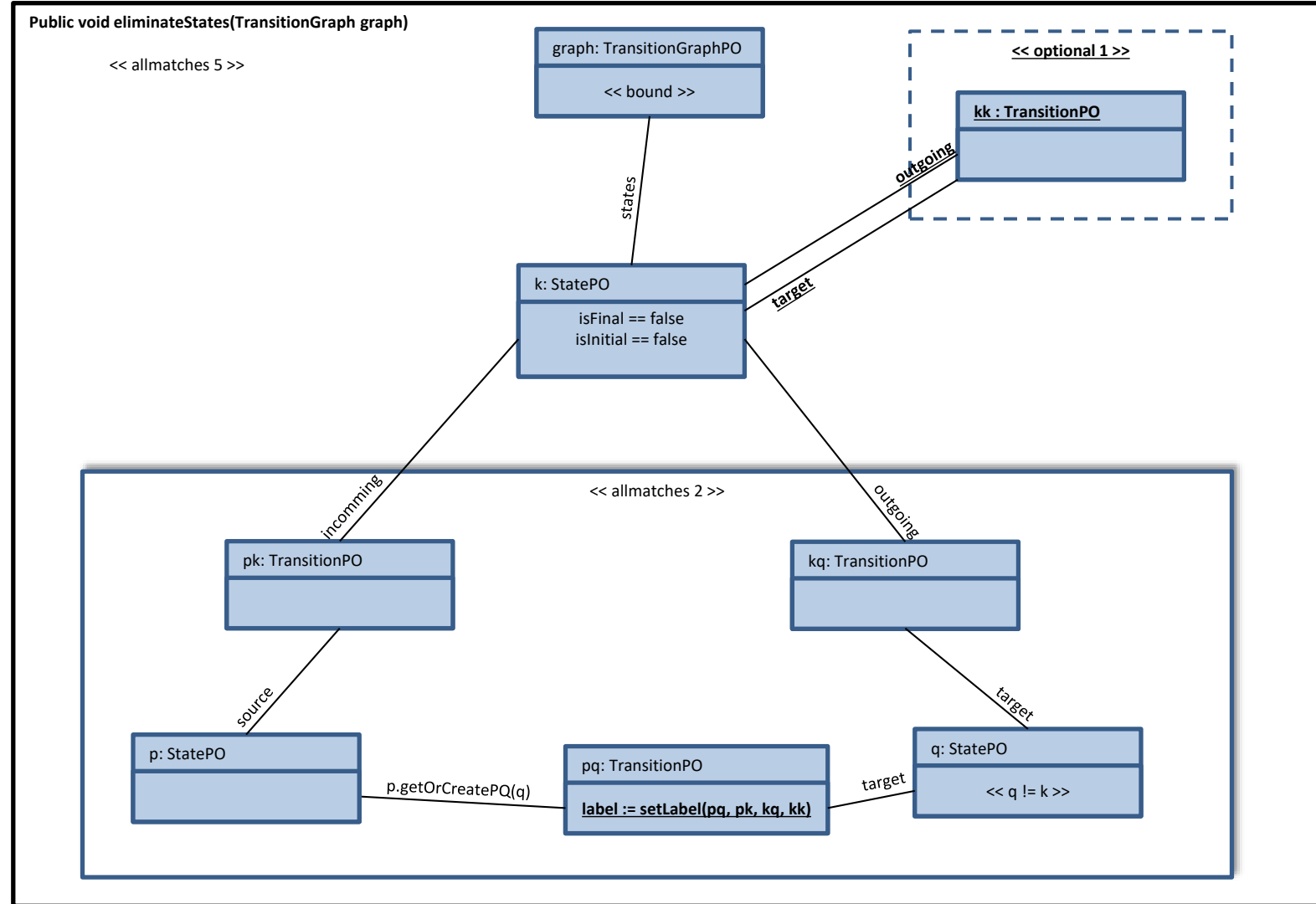
newTransitions.add(pq)





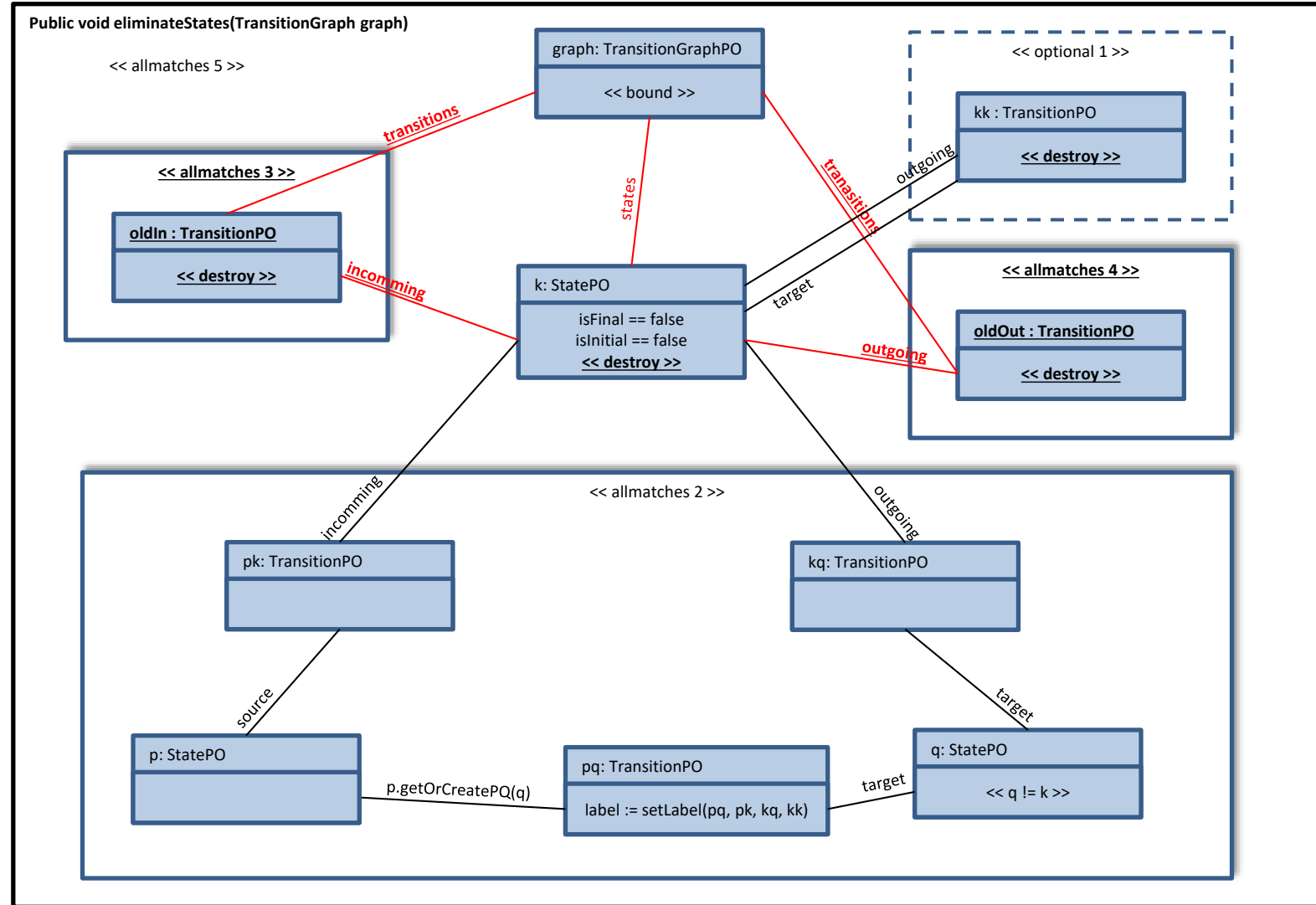
# Pattern

$pq \leftarrow \text{getExpressionForElimination}(k, p, q, A)$



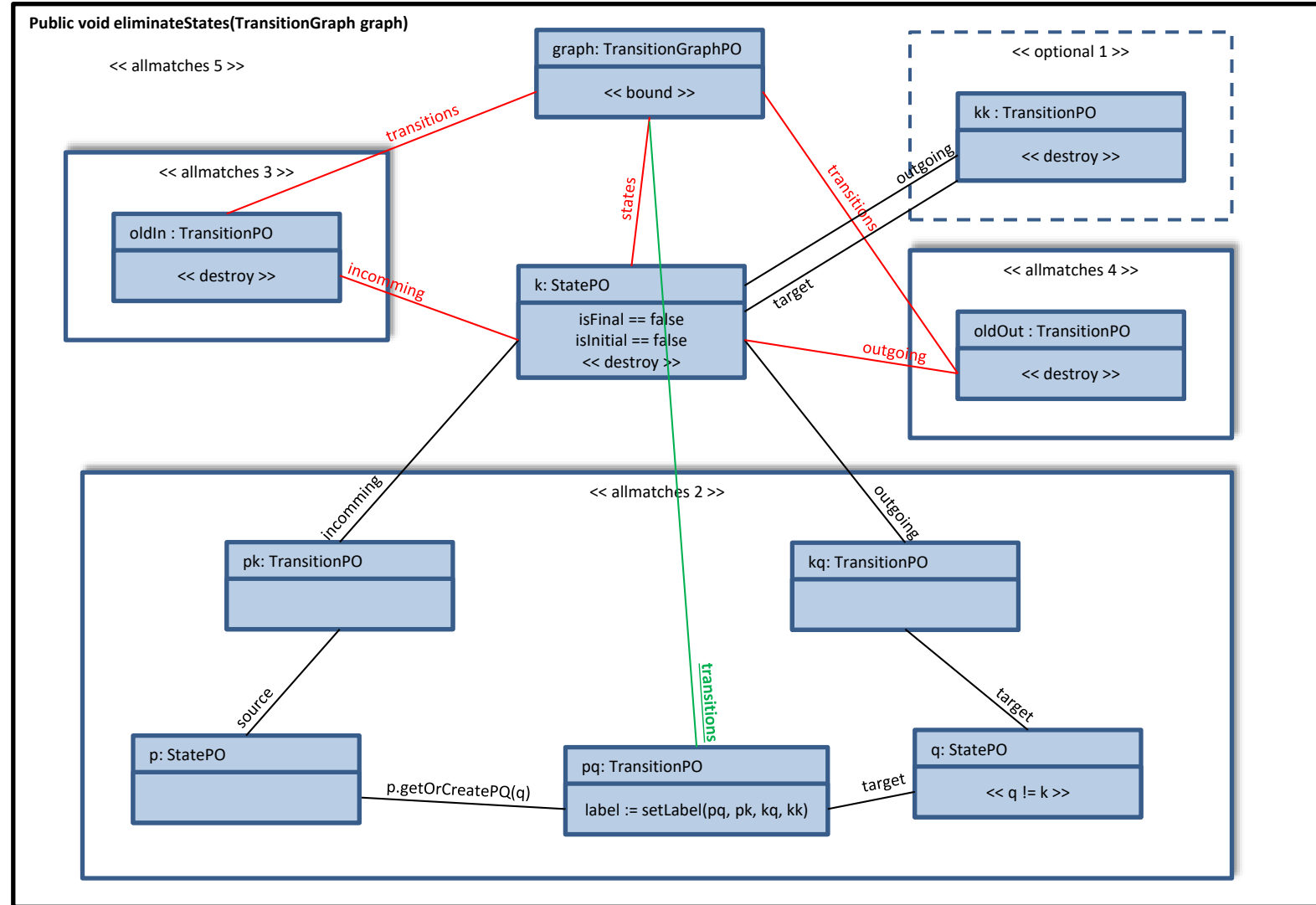
# Pattern

remove k and all its incoming and outgoing transitions from A



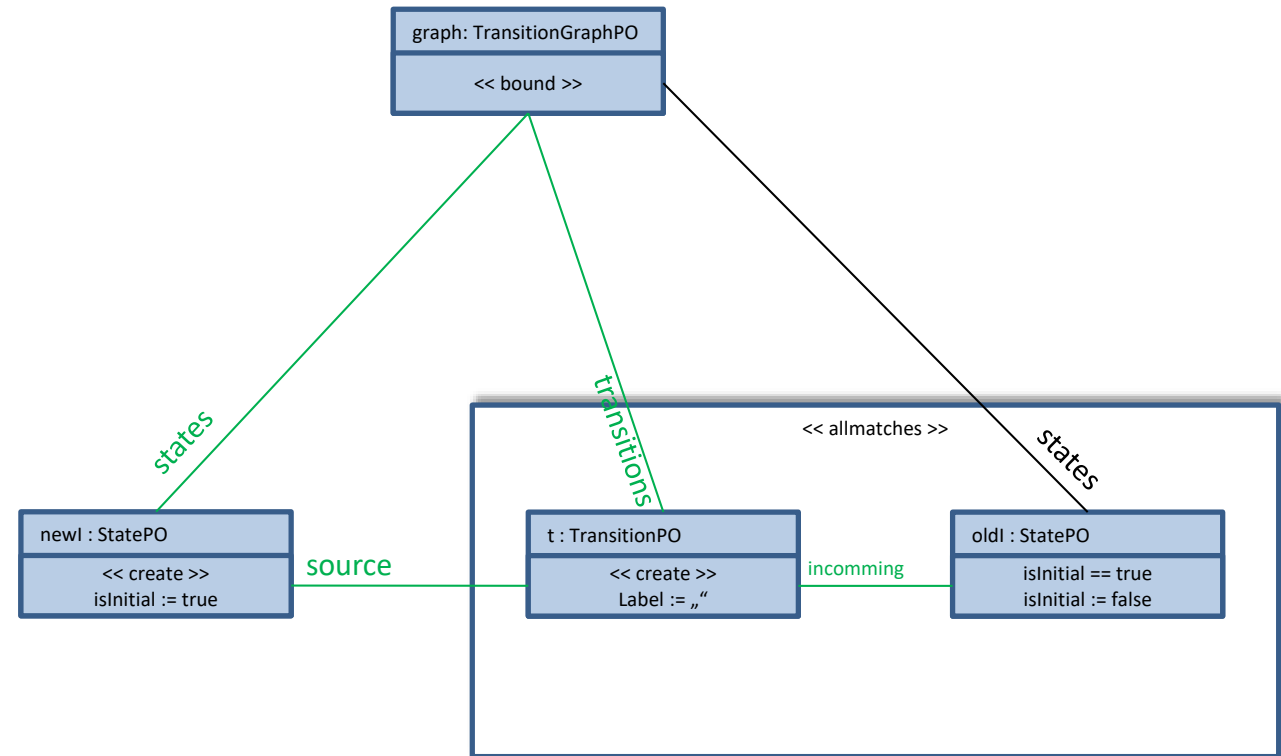
# Pattern

A.add(newTransitions)



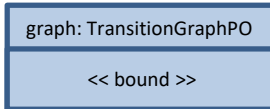
# Extension 1

Public void uniformInitial (TransitionGraph graph)



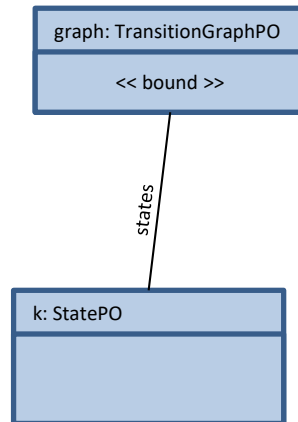
# Code

```
TransitionGraphPO graph =  
    new TransitionGraphPO( graph )
```



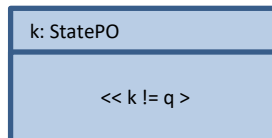
# Code

StatePO k = graph.createStatesPO()



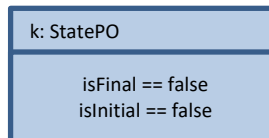
# Code

k.hasMatchOtherThen(q)



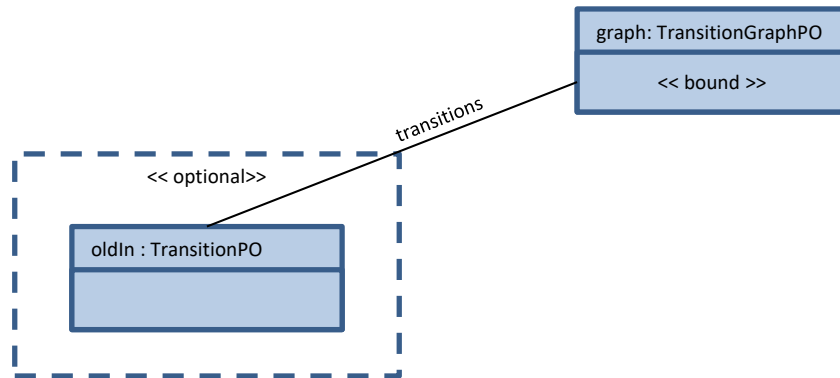
# Code

```
k.createIsFinalCondition(false)  
    .createIsInitialCondition(false);
```





# Code

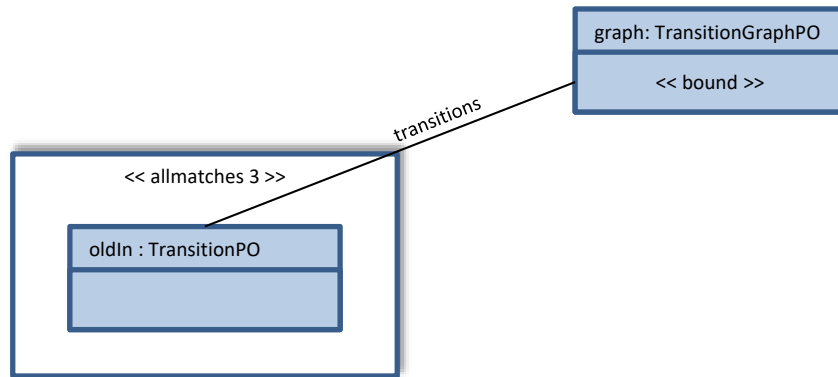


```
graph.createSubPattern();
```

```
TransitionPO oldIn = graph.createTransitionsPO();
```

```
graph.endSubPattern();
```

# Code



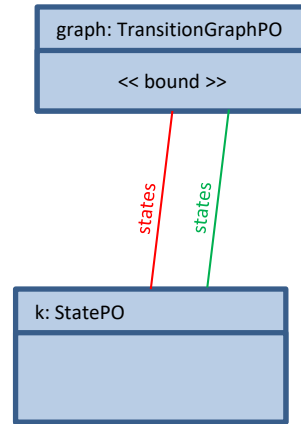
```
graph.startSubPattern();
```

```
graph.getOnDutyPattern().withDoAllMatches( true );
```

```
TransitionPO oldIn = graph.createTransitionsPO();
```

```
graph.endSubPattern();
```

# Code

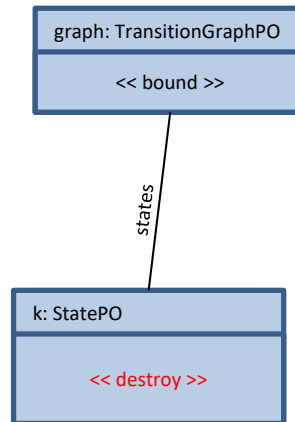


```

graph = graph.createStatesPO(Pattern.DESTROY);
graph = graph.createStatesPO(Pattern.CREATE);
    
```

# Code

k.destroy();



# Performance

	JFLAP	SDMLib	SDMLib (+Evaluation)
Leader3_2	0.09	0.040	0.078
Leader4_2	0,14	0.044	0.215
Leader3_3	0,49	0.062	0.417
Leader5_2	3,46	0.335	2.573
Leader3_4	4,37	0.155	1.090
Leader3_5	58,6	0.302	1.333
leader4_3	57.78	5.696	34.009
leader6_2	143.12	1.874	22.992
leader3_6	461.64	0.548	1.304

# Scaling

- **Problem: Strings get exponentially bigger:**
  - Bigger examples are too big for memory...
    - Using fileoperations for strings, that are too big

# State Elimination - The SDMLib Solution

Alexander Weidt & Albert Zündorf

University Kassel – Software Engineering Research Group